

Секция информационных технологий

Таким образом, было создано пять таблиц. Все значения полей таблиц являются неделимыми, не имеется совпадающих записей. Все неключевые поля в таблицах зависят только от ключа таблицы и не зависят друг от друга. Это значит, что все таблицы БД приведены к 3 нормальной форме.

В каждой таблице имеется первичный ключ (*primary key*), который является идентификатором объекта. Каждый идентификатор – это счетчик (*identity(1,1)*), который автоматически увеличивается на единицу при добавлении в таблицу новой записи. Таблицы также включают поля с символьным типом данных (*nvarchar*). Для установления связей между данными в таблицах были созданы внешние ключи (*foreing key*). Все поля таблицы должны быть обязательно заполнены (значение *NOT NULL*).

ЛИТЕРАТУРА

1. Пинкус, М. Введение в базы данных [Электронный ресурс] / М. Пинкус. – Режим доступа: <http://onmcsso.narod.ru/db/>.
2. Карпова, И. П. Проектирование реляционных баз данных [Электронный ресурс] / И. П. Карпова. – Режим доступа: <http://rema44.ru/resurs/study/dbprj/dbprj.html>.
3. Эталонная модель управления данными: ГОСТ Р ИСО/МЭК ТО 10032-2007. Введен 27.12.07. – Москва: Стандартинформ, 2009. – с.3.
4. Ицик, Б. Г. Microsoft SQL Server 2012. Основы T-SQL / Б. Г. Ицик. – М. : Эксмо, 2015. – 400 с.

УДК 378.147

Студ. А. В. Бабако

Науч.рук.: доц., к.т.н. Н. Н. Пустовалова
(кафедра информационных систем и технологий, БГТУ),
доц., к.п.н. Н. П. Коровкина

(кафедра автоматизации производственных процессов и электротехники, БГТУ)

**ОБУЧАЮЩАЯ ПРОГРАММА ДЛЯ ИЗУЧЕНИЯ ФАЙЛОВОГО
ВВОДА/ВЫВОДА НА ЯЗЫКЕ C/C++**

Приложение для изучения файлового ввода/вывода на языке C/C++ реализовано в виде двух этапов обучения. Первый этап представлен в виде теста, на втором этапе пользователь должен отвечать на различные вопросы по теме. Для перехода с первого этапа на следующий необходимо правильно выполнить больше 70%

заданий, в противном случае этап будет не доступен. Процент выполненных заданий отображается на главной форме (Меню). Если выполненных правильно заданий недостаточно, пользователь должен пройти тест заново.

Знание темы «Файлового ввода/вывода на языке C/C++» является фундаментом работы с информацией. Обмен информацией между программой и внешними устройствами осуществляется с помощью операций ввода/вывода. Под вводом/выводом в программировании понимается процесс обмена информацией между оперативной памятью и внешними устройствами: клавиатурой, дисплеем, магнитными накопителями и т. п. Ввод – это занесение информации с внешних устройств в оперативную память, а вывод – вынос информации из оперативной памяти на внешние устройства. Такие устройства, как дисплей и принтер, предназначены только для вывода данных; клавиатура – устройство ввода информации. Магнитные накопители (диски, ленты) используются как для ввода, так и для вывода.

Основным понятием, связанным с информацией на внешних устройствах компьютеров, является понятие файла. Всякая операция ввода-вывода трактуется как операция обмена с файлами: ввод – это чтение из файла в оперативную память; вывод – запись информации из оперативной памяти в файл. Поэтому вопрос об организации в языке программирования ввода-вывода сводится к вопросу об организации работы с файлами.

Для изучения темы в окно формы загружается html-документ с краткой информацией по работе с файлами, разбитый на логические блоки: создание потока, связь с файлом; открытие файла, режимы доступа; обмен информацией.

Подробно рассматривается организация записи информации в файл сначала на языке C++, а именно: вставка в поток (операция >>); посимвольная запись; метод `fl.write(const char* buf, type num)`, где `fl` – поток, связанный с файлом, первый аргумент – записываемая информация, второй – размер.

Аналогично с записью рассматривается считывание информации из файла: извлечение из потока (операция <<); посимвольное считывание; метод `read(char* buf, type num)` для работы со структурами, массивами.

При открытии файла, курсор устанавливается в начале либо в конце файла. Для работы с информацией необходимо иметь быстрый

доступ в любую позицию файла, поэтому рассматриваются способы использования функций позиционирования.

Закончив работу с файлом, первое, что необходимо сделать – отсоединить поток от файла, разорвать связь посредством функции `close()`.

После ознакомления с теоретическим материалом пользователю предоставляется возможность закрепить его и проверить, насколько хорошо усвоен материал, пройдя этап, включающий, как и запись ответа в маскированное поле, так и тестовые задания (рис. 1).

Чем характеризуется данный режим открытия файла? `ios::ate`

- ☐ Установить указатель позиционир. фай-ла на конец
- ☐ начать поиск с конца файла;
- ☐ Открыть файл для дозаписи в его конец

Параметр `origin` определён как константа в классе `ios`, поск от текущей позиции в файле

- ☐ `ios::beg`
- ☐ `ios::cur`
- ☐ `ios::end`

Для чтения строки из файла используется функция (), которая является функцией класса `ifstream`.

Функция () считывает символ из файла. Функция () записывает символ в файл.

Continue Exit

Рисунок 1 – Окно для проверки знаний студентов

Поскольку в настоящее время для работы с файлами используется также и язык С, то с помощью данного обучающего программного средства изучаются способы обмена данными на этом языке.

При этом последовательность действий и их изучение при вводе/выводе сохраняется:

- создается специальная структура с именем `FILE`;
- открывается файл с помощью функции `fopen()`, где прописывается имя файла и метод открытия файла, который позволяет предусмотреть ситуацию, существует ли файл и установка курсора;
- работа с информацией осуществляется посимвольно, по строкам, по блокам;

– позиционирование, осуществляется посредством параметров: SEEK_SET, SEEK_CUR, SEEK_END;

– в конце работы с файлом «отвязывается» структура FILE от файла.

Приложение реализовано на языке C# с привязкой html-документов и ориентировано на студентов, изучающих язык программирования C/C++. Оно помогает освоить теоретические основы, которые являются фундаментом практики.

УДК 625.539.3

Студ. И.Д. Сафончик

Науч. рук. доц. Н.В. Пацей

(кафедра информационных систем и технологий, БГТУ)

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ УПРАВЛЕНИЯ ЕВРОПЕЙСКОЙ СИСТЕМОЙ ПЕРЕВОДА И НАКОПЛЕНИЯ УЧЕБНЫХ КРЕДИТОВ (ECTS) ДЛЯ IT СПЕЦИАЛЬНОСТЕЙ

Европейская система перевода и накопления кредитов была первоначально основана в 1989 году как пилотная схема в рамках программы Эразмус. В то время ее целью было обеспечить признание периодов обучения студентов, учащихся за границей путем перевода кредитов. В качестве трансферной системы ECTS применяется более, чем в 30 странах и используется более, чем в одной тысячи учреждений высшего образования.

40 государств, участников Болонского процесса определили ECTS как один из краеугольных камней высшего образования в Европе. Большое количество стран официально применяет ECTS как накопительную систему для своих собственных систем высшего образования, а другие страны готовятся к ее применению. В некоторых странах ECTS стала необходимой для получения аккредитации. Цюрихская конференция по переводу и накоплению кредитов, проведенная в октябре 2002 года Европейской ассоциацией вузов, подтвердила центральную роль ECTS в высшем образовании. ECTS будет иметь фундаментальное место в разработке национальной и Европейской квалификационной структуры. Схемы, основанные на ECTS начинают применяться и на других континентах.

Правильное использование ECTS зависит от усилий тысяч высоко мотивированных людей, занятых в высшем образовании в разных условиях и культурном окружении.